Implementation and Applications of CORDIC Algorithm in Satellite Communication

Satish Sharma[1], Sunil Kulkarni[2] and P.Lakshminarsimahan[3]
Digital Systems Group, ISRO Satellite Centre, Airport Road, Banagalore-560017, India[1]
Digital Systems Group, ISRO Satellite Centre, Airport Road, Banagalore-560017, India[2]
Digital Systems Group, ISRO Satellite Centre, Airport Road, Banagalore-560017, India[3]

Email: {satish@isac.gov.in[1], kulkarni@isac.gov.in[2], vihariam@isac.gov.in[3]}

***Abstract*— CORDIC stands for COrdinate Rotation DIgital Computer, is a versatile algorithm, widely used for VLSI implementation of digital signal processing (DSP) based communication systems. This paper presents the use of CORDIC in satellite communication systems. Specifically, Digital Down Converter (DDC), Direct Digital Frequency Synthesizer (DDFS), and use of CORDIC in digital modulator and demodulator has been explained. The algorithm and its applications are first realized on System Vue software tool and then designs are coded in VHDL. Apart from the implementation aspects, paper presents new results regarding the number of branches required for computation. The paper also analyzes the performance of CORDIC based DDC and DDFS on the basis of hardware requirement. The system is implemented on XILINX FPGA platform and tested in real time for its performance**

***Index Terms*— Digital down convertor, Frequency synthesizer, Signal processing, Satellite communication.**

I. INTRODUCTION

Satellite communication systems, now days going through a series of rapid changes in terms of modulation techniques and implementation. More & more DSP algorithms are used to increase the performance of the system. Thus, there is an increasing need for the efficient implementations of complex arithmetic operation require to, realize these algorithms. Among the hardware-efficient algorithms, is a class of iterative solutions for trigonometric and transcendental functions that use only shifts and adds to perform. This algorithm is called CORDIC, was first introduced by Jack E Volder in 1959. CORDIC is an arithmetic algorithm widely used in the computation of elementary functions and digital signal processing applications, particularly where large amount of rotation operations are necessary. It calculates the trigonometric functions of sine, cosine, arctangent, magnitude and phase, to any desired precision. It can also calculate hyperbolic functions. Apart from these basic realization CORDIC can be used to implement various transform like Discrete Cosine Transform (DCT), Hilbert transform and Discrete Fourier transform (DFT).

CORDIC works on the basis of iterative rotation of a two-dimensional vector using only add and shift operations. The DSP based communication system implementation using CORDIC is more hardware efficient than conventional as CORDIC doesn't require any multiplier and accumulator. With the digital realization of the system, we can optimize the communication system at algorithm level plus at implementation level.

The basic theory of CORDIC is explained in section II. Section III deals with the error analysis and data width requirements in CORDIC implementation followed by in depth analysis of CORDIC based DDFS. In section V, issues related to FPGA implementation of CORDIC based DDFS and results of real time testing are described. The use of CORDIC based DDFS in other system is dealt in section VI. Section VII explains about CORDIC based DDC, its implementation and real time spectrum followed by conclusion at the end.

II. CORDIC THEORY

In its most general form [1], the CORDIC iterative equations can be described as

$$X_{i+1} = X_i - m.Y_i.d_i.2^{-i} \quad (1)$$

$$Y_{i+1} = Y_i + X_i.d_i.2^{-i} \quad (2)$$

$$Z_{i+1} = Z_i - d_i\alpha_i \quad (3)$$

where d_i = -1 if $Z_i < 0$, +1 otherwise and

$\alpha = \tan^{-1}(2^{-i})$

$$m = \begin{cases} 1 & \text{for circular co-ordinate system} \\ 0 & \text{for linear co-ordinate system} \\ -1 & \text{for hyperbolic co-ordinate system} \end{cases}$$

Variables X and Y represent the vectors and Z is the angle accumulator variable, which keeps track of the angle that already

been rotated. Variable d, decides the direction of next rotation and the m specifies the working co-ordinate system. α_i is the angle that the vector will be rotated by during the i^{th} iteration.

A. Modes of operation

CORDIC operates in two mode vectoring mode and rotation mode. In vectoring mode the initial vector is rotated to align it along X-axis. This mode is used for calculation of Arc tan. Input is X and Y variable and output comes on Z.

In rotation mode, the angle accumulator is initialized with the desired rotation angle. The rotation decision at, each iteration is made to diminish the magnitude of the residual angle in the angle accumulator. The decision is therefore based on the sign of the residual angle after each step. The equation (1), (2), and (3) provide this result

$$x_n = A_n[x_0 \cos z_0 - y_0 \sin z_0] \quad (5)$$

$$y_n = A_n[y_0 \cos z_0 + x_0 \sin z_0] \quad (6)$$

$$z_n = 0 \quad (7)$$

$$A_n = \prod_n \sqrt{1+2^{-2i}} \quad (8)$$

Where A_n is CORDIC gain.

B. CORDIC Architecture

Based on the implementation of equations (1) , (2) ,(3) , there are two architectures [2], serial and parallel. In our design we have chosen parallel implementation, because of the requirement of higher speed which is supported by the parallel architecture with penalty of more hardware. Serial architecture has an advantage in terms of hardware efficiency but it cannot work at higher speed. In parallel architecture each, iteration has one CORDIC unit. One CORDIC unit structure is shown in fig.1.

III. Error Analysis, Choosing Number of Iterations and Data Path Width

In applying circular-mode CORDIC to any application, it is crucial to understand how seemingly independent parameters can affect each other. First, it is necessary to determine how much iterations are necessary to properly resolve a radian angle in *b* fractional bits. Assuming that MSB bit is for sign. The unit in the least precision (ulp) 2^{-b} in the angle representation is meaningful only when it can be resolved by the iteration. That is, at the final iteration where i = n-1, and n is number of iterations. α_{n-1} should be less than or equal to 2^{-b} . This ensures that the final error in angle is less than 2^{-b}. The value $\alpha_i = \arctan 2^i$ can be

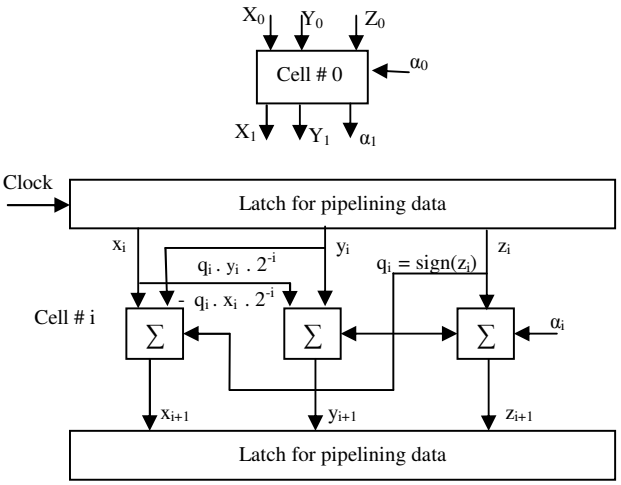

Fig. 1. Parallel CORDIC architecture.

approximated as 2^{-i} for large values of i. Therefore, $2^{-(n-1)}$ should be less than or equal to 2^{-b}. So, the minimum n is b+1, which is the minimum number of iterations to correctly resolve the given angle. In most DDFS applications, however, the angle is given in a normalized format such that $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ is scaled to $\left(-\frac{1}{2}, \frac{1}{2}\right)$ because angle accumulator overflow is awkward to deal with in the radian format. When the normalized format is used, the angle table should also contain normalized entries. Suppose the angle is given by *b* fractional bits. Then, the normalized angle's *ulp* is 2^{-b} and it should be resolved by α_{n-1}.

$$\alpha_{n-1} \le 2^{-b}$$

But,

$$\alpha_{n-1} = \frac{\arctan\left(2^{-(n-1)}\right)}{\pi} \approx \frac{2^{-(n-1)}}{\pi}$$

Therefore, $\frac{2^{-(n-1)}}{\pi} \le 2^{-b}$

This means that the minimum n is b. Finally, if n iterations are to be made, the data path width *w* should be at least n – 1 (fractional). This is because, for n iterations, the largest shift is n – 1, and only when $w \ge$ n – 1, can the largest shift contribute to the iteration.

IV. CORDIC Based DDFS

In satellite communication systems, frequency synthesizer finds application in demodulator, in PLL loop and in modulator for generating sine & cosine carrier wave. Conventionally Digital controlled oscillator (DCO) followed by the ROM look up table are used to generate the sine / cosine wave form. The major difficulty with this approach is that the size of the ROM table

increases exponentially with the width (i.e., the resolution) of the output. Even though there have been several techniques proposed to mitigate this problem, which basically partition the ROM table to reduce the size, the exponential relationship still persists and a resolution higher than 12 bits is currently considered impractical. Using an arithmetic unit, is an alternative approach that has been getting attention recently. Especially the CORDIC algorithm which has been dominant in this approach because it only requires shift-and-add operation while other approaches based on functional approximations generally involve more complicated arithmetic operation such as multiplication. The general architecture of DDFS with CORDIC /ROM look up table is shown in fig.2. The frequency output f_{DCO}, system clock f_{SYS}, frequency control word *K*, and number of phase accumulator bits *N,* are related as [3]

$$f_{DCO} = \frac{f_{SYS} * K}{2^N} \quad (9)$$

The CORDIC algorithm is used in rotation mode for calculation of sin and cosine angle. The input to the CORDIC equation is the angle for which sine and cosine has to be calculated. Setting y component of the input vector to zero in equation (5) and (6) reduces the rotation mode to

$$x_n = A_n.x_0.\cos z_0$$

$$y_n = A_n.x_0.\sin z_0$$

By setting x_0 equal to $1 / A_n$, the final value of x and y gives the sin and cosine angle of the input angle.

V. FPGA IMPLEMENTATION OF CORDIC BASED DDFS

Before implementation the DDFS design is simulated in system vue software. We have design the system for 14 bit, so
final error (in angle) $\leq 2^{-b} \leq 2^{-13} \leq 0.0001220703125$;
For 'n' bits of Phase accumulator Maximum multiplicand will be 2^{n-2} for amplitude because if the multiplicand value is more than this then $\tan^{-1}(2^{-0}) = 45$ which the highest angle to be represented

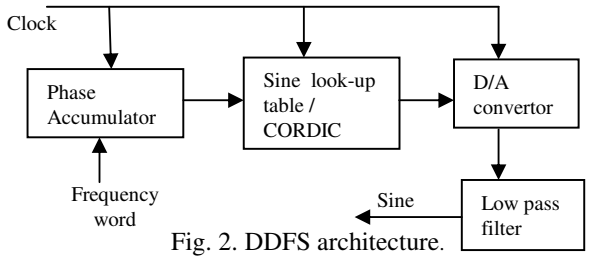

Fig. 2. DDFS architecture.

cannot be accommodated in 'n' bits. Therefore, maximum number of iterations which it can go up to will be (n-2) not 'n'. We have design and simulated the DDFS for (n-2) iterations and 'n' iterations but we didn't find any significant improvement in the output. Beyond (n-2) iterations the value of $\tan^{-1}(2^{-i})$ will be 0 in signed integer format.

Angle Mapping is done to map each degree to its corresponding value according to the number of bits taken. Therefore if 180° represents = 2^{n-1}, Then 1° will be equal to (2^{n-1} / 180°). Hence the angle must be multiplied by (2^{n-1} / 180°). We also used initial $\pm\pi/2$ rotation to increase the range of the CORDIC algorithm. The system vue output for 1 KHZ frequency is shown in fig.3

After adjusting the parameters, the design is coded in the VHDL and simulated in Model sim software. Finally for real time testing the design is targeted to the XILINX (xc2v3000-4fg676) FPGA. Table-1 shows the comparison of hardware requirements for ROM look up table. Fig. 4 shows the real time captured wave form on oscilloscope

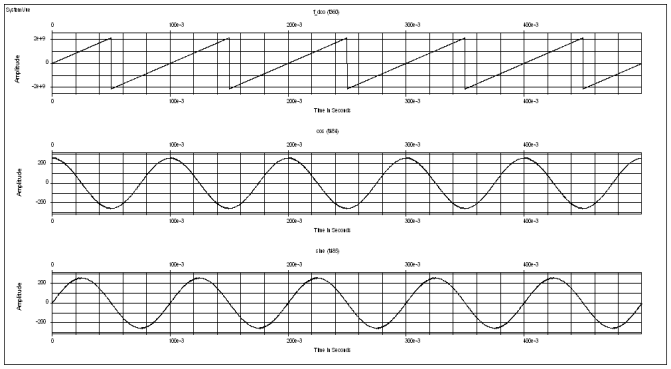

Fig. 3. Simulated phase accumulator output with Sine & Cosine waveform of 1 KHz frequency.

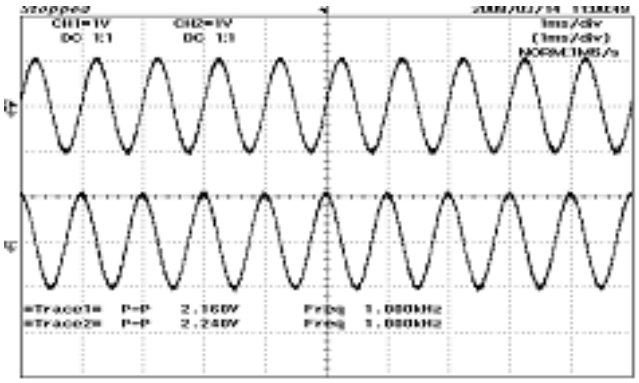

Fig. 4. Real time Sin & Cosine waveform at 1 KHz frequency

TABLE -I
HARDWARE REQUIREMENT COMPARISON OF CORDIC BASED AND ROM LOOK UP TABLE BASED DDFS

Serial No.	DDFS Implementation in Xilinx (XCV1000-4bg676)		
	Phase Accumulator Bits & Phase Resolution	*Slices used for CORDIC based DDFS*	*Slices used for ROM based DDFS*
1	Bits: 10 Phase Resolution : 0.3515625	230/14336 (1%)	179/14336 (1%)
2	Bits: 14 Phase Resolution : 0.02197	540/14336 (3%)	10320/14336 (72%)
3	Bits: 24 Phase Resolution : 0.000021	841/14336 (5%)	Not possible

.

From Table-I, it is clear that with CORDIC based DDFS we can get more phase resolution in less hardware. The hardware requirement for look up table based DDFS increase exponentially. Up to 10 bit phase resolution lookup table approach is better but beyond that CORDIC outscore in terms of hardware requirement. The important performance parameter for DDFS is spurious free dynamic range (SFDR). The SFDR is ratio of amplitude of the desired frequency component to that of the largest undesired frequency component in the output of DDFS. It is expressed as

$$SFDR = 20\log_{10}\left(\frac{A_p}{A_s}\right)$$

Where A_p is the amplitude of the desired frequency component and A_s is that of the undesired component. The SFDR can be calculated approximately by examine the spectrum of generated waveform. The fig. 5 shows the real time spectrum of 1 MHz sine wave. It is clear that SFDR is approximately 80 db.

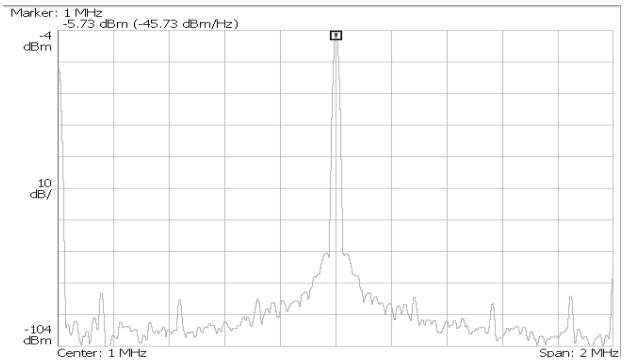

Fig. 5. Real time spectrum of 1 MHz sine wave generated using CORDIC based DDFS

VI. APPLICATIONS OF CORDIC IN SATELLITE SUB SYSTEMS

The DDFS is one of the requirements in digital modem design. Particular to our satellite system, we are using BPSK modulation technique, so at transmitter end DDFS can be used in BPSK modulator as shown in fig.6. Input data is used to select the phase and that is added with the phase of the carrier. Finally phase information is converted in to amplitude by CORDIC. D/A is used to get the analog output. The system can be upgraded to M-PSK by suitable selection of phases. At receiver the CORDIC based DDFS finds application in Costas loop demodulator & Bit synchronizer as shown in fig. 7 & 8.

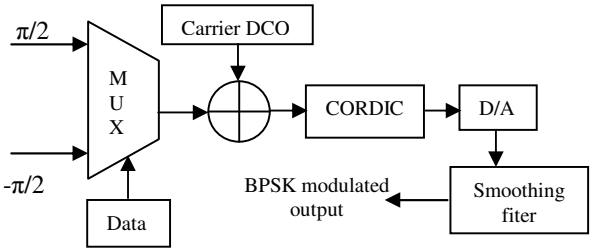

Fig. 6 Digital BPSK modulator

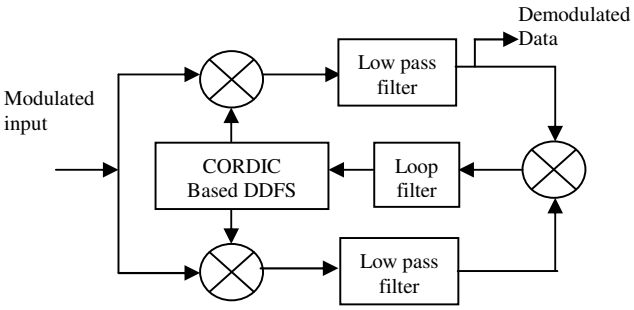

Fig. 7 Costas loop demodulator

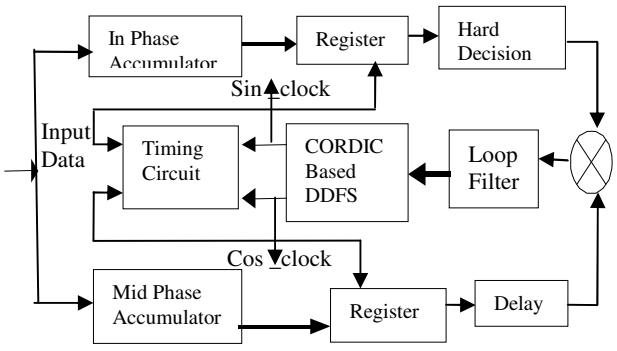

Fig. 8 In phase / Mid phase bit synchronizer

VII. CORDIC BASED DDC FOR SATELLITE RECEIVER

DDC is one of the important modules of the any receiver system. There are various algorithm to down convert the RF input to any low IF frequency. One of the simple methods is the use of multiplier. For I-Q demodulation we require two multiplier and a DDFS as shown in fig.9.

Instead of using two multiplier & look-up table based DDFS, CORDIC can be used directly to down convert [4] and generate I and Q component of the received input as shown in fig.10.

A proto DDC using CORDIC, having 14 iteration and 12 bit amplitude resolution, is realized in XILINX FPGA. X input of CORDIC is fixed at zero and at Y, signal to be down converted is given [5].

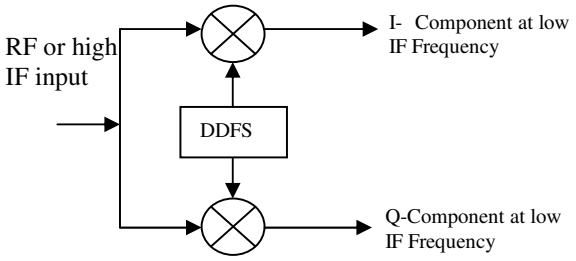

Fig. 9 Multiplier based down conversion

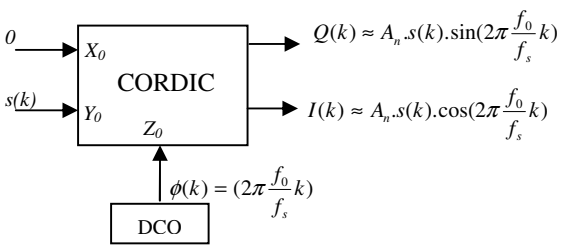

Fig. 10 CORDIC based down conversion

DCO is operated on 1 MHz with sampling clock of 20 MHz the design is simulated, systhensized and real time tested on the XILINX platform by giving a 3 MHz sine wave input and observing output in spectrum analyzer. The SFDR achieved in this design is apprimately 72 dB. The design can go up to 215 MHz. Fig.11 shows the spectrum of down converted signal at 2 & 4 MHz.

VIII. CONCLUSION

CORDIC is very versatile algorithm and just by changing few parameters we can realize many useful function. Specifically in this paper, applications of CORDIC algorithm in satellite communication system are analyzed. The CORDIC based DDFS

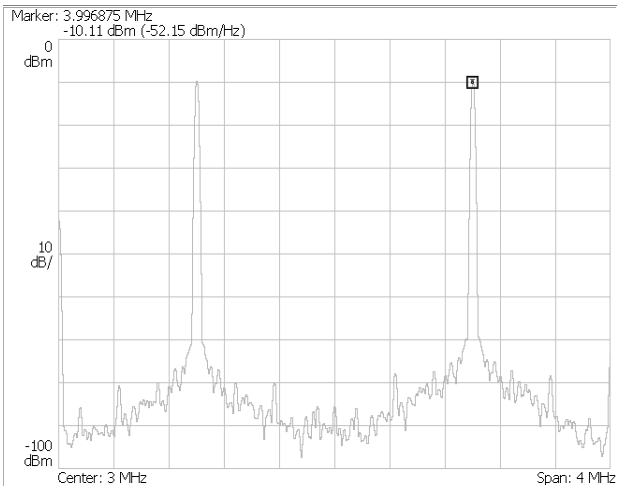

Fig.11 Down converted output at 2 MHz & 4 MHz.

is dealt in depth from concept to FPGA realization. The DDFS is realized on XILINX platform and tested real time. Spectrum of DDFS output shows a SFDR of 80dB. It was found that over 10 bit phase resolution CORDIC based DDFS has advantage in terms of hardware on ROM based DDFS. We have found out that in case of DDFS "n-2" iterations are enough instead of 'n'. Paper also explained about CORDIC based DDC. SFDR in case of DDC is 72 dB. Apart from these, applications of CORDIC in digital modulator, demodulator and bit synchronizer are also presented. We have chosen parallel architecture for CORDIC algorithm implementation for all application as it provides advantage in terms of speed of operation. The various applications of CORDIC are simulated, coded in VHDL and real time tested on XILINX platform.

ACKNOWLEDGMENT

Authors would like to thanks Maduri Rao, Shruthi H.A, Vanshri R and Vidhya T.P for carrying out the simulations at various level of design & helping in real time testing of design.

REFERENCES

[1] R. Andraka " A Survey of CORDIC Algorithm for FPGA Based Computers" *Proc. of ACM/SIGDA Sixth International Symposium on FPGAs*, pp 191-200, Feb.1998, Monterrey, CA.

[2] Herbert Dawid and Heinrich Meyr, "*Chapter 24 ,Digital Signal Processing for Multimedia Systems*", Meyr Synopsys Inc.

[3] Bar-Giora Goldberg ,*Digital Frequency Synthesis Demystified : DDS and Fractional-N-PLLS*, Eagle Rock,VA, LLH Technology,1999

[4] Javier valls, Trini Sansaloni, and A.P.Pascual "*The Use of CORDIC in Software Defined Radios: A tutorial*" IEEE Communications Magazine, pp 46-50, Sept.2006.

[5] M. Loehning, T. Hentschel, and G. Fettweis, "*Digital Down Conversion in Software Radio Terminals*", in 10th European Signal Processing Conference, vol.3 ,Sept 2000 Tampere, Finland.